Mission Thread Workshop -

Lessons Learned in Legacy System Architecture Evaluation Using Mission Threads as a Starting Point

SATURN Conference 2012

Mike Gagliardi, Bill Wood

This material is based upon work funded and supported by the Department of Defense under Contract
No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute,
a federally funded research and development center.

Problem

Integration and operational problems arise due to inconsistencies, ambiguities, and omissions in addressing quality attributes between system and software architectures. This is further exacerbated in an SoS.

Example quality attributes: predictability in performance, security, availability/reliability, usability, testability, safety, interoperability, maintainability, force modularity, spectrum management.

Functionality and capability are critically important, but the architecture must be driven by the quality attributes. Specifying and addressing quality attributes early and evaluating the architecture to identify risks is key to success.

The Need for Augmented End-to-End Mission Threads in DoD SoS Architecture Definition

DoDAF provides a good set of architectural views for an SoS architecture. However, it inadequately addresses cross-cutting quality attribute considerations.

System use cases focus on a functional slice of the system.

More than DoDAF and system use cases are needed to ensure that the SoS architecture satisfies its cross-cutting quality attribute needs.

SoS end-to-end mission threads augmented with quality attribute considerations are needed to help define the SoS Architecture and then later evaluate the SoS architecture and constituent system/software architectures.

Definitions (DoD Context)

Vignette: A description of the geography, own force structure and mission, strategies and tactics, the enemy forces and their attack strategies and tactics, including timing. There may be associated Measures of Performance (MOP) and Measures of Effectiveness (MOE). A vignette provides context for one or more *mission threads*.

Mission Thread: A sequence of end-to-end activities and events beginning with an opportunity to detect a threat or element that ought to be attacked and ending with a commander's assessment of damage after an attack. C4ISR for Future Naval Strike (Operational)

Sustainment: A sequence of activities and events which focus on installation, deployment, logistics and maintenance.

Development: A sequence of activities and events that focus on re-using or re-engineering legacy systems and new adding capabilities

Acquisition: A sequence of activities and events that focus on the acquisition of elements of an SoS, and the associated contracts and governance

Vignettes Are the Starting Point – Example Wording

Two ships (Alpha and Beta) are assigned to integrated air and missile defense (IAMD) to protect a fleet containing two high-value assets (HVA). A surveillance aircraft SA and 4 UAVs are assigned to the fleet and controlled by the ships. Two UAVs flying as a constellation can provide fire-control quality tracks directly to the two ships. A three-pronged attack on the fleet occurs:

- 20 land-based ballistic missiles from the east
- 5 minutes later from 5 aircraft-launched missiles from the south
- 3 minutes later from 7 submarine-launched missiles from the west.

The fleet is protected with no battle damage.

Mission Threads Flow from Vignettes – Example (Non-Augmented)

1. 20 land-based missiles launched - X minute window

2. Satellite detects missiles - cues CMDR

3. CMDR executes re-planning – reassigns Alpha and Beta

4. Satellite sends track/target data - before they cross horizon

5. Ships’ radars are focused on horizon crossing points

…

N Engagement cycle is started on each ship

N+1. Aircraft are detected heading for fleet

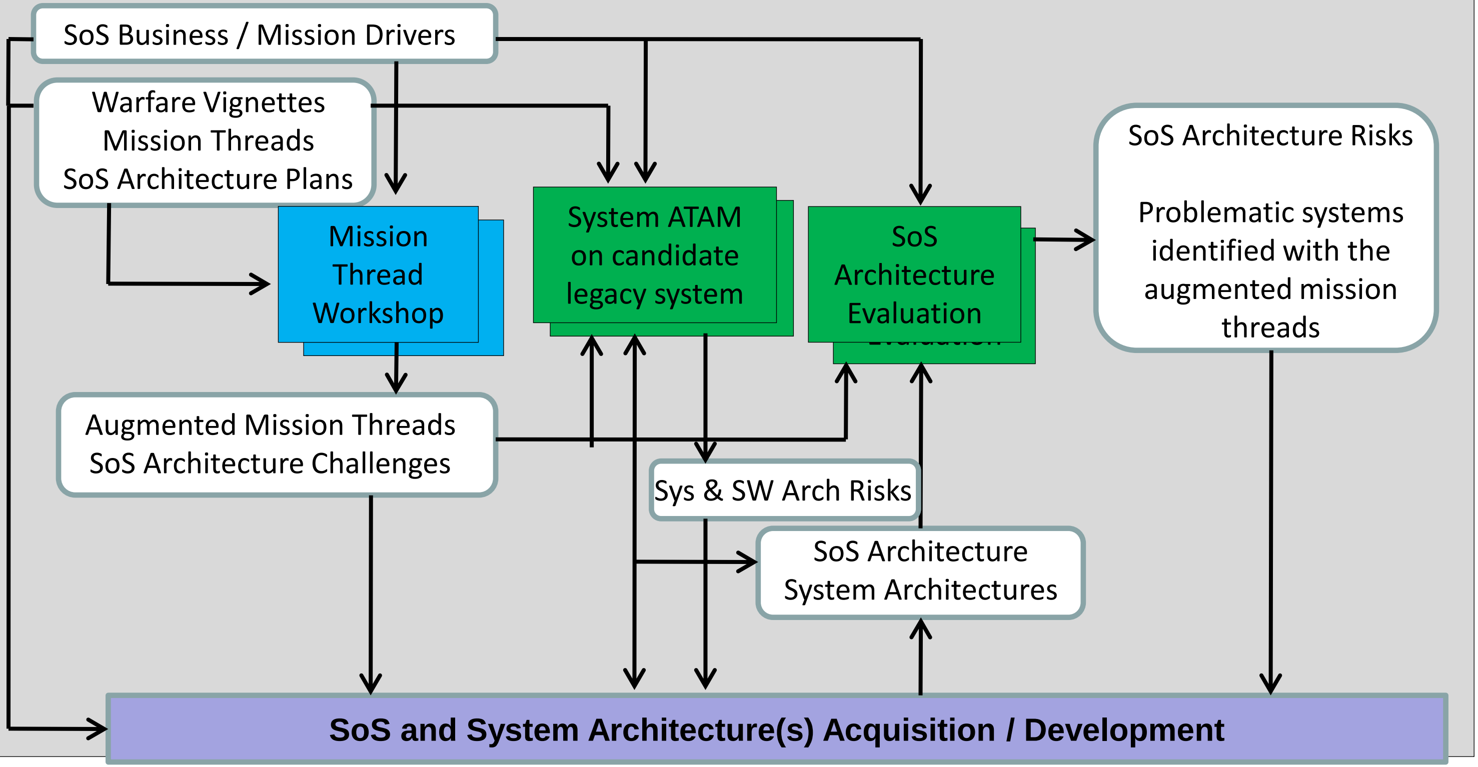
N+2. SA detects missile launches – tells CMDR

N+3. CMDR does re-planning - UAVs are re-directed

N+4. FCQ tracks are developed from UAV inputs

SoS Architecture Quality Attribute Specification and Evaluation Approach

- Early elicitation of quality attribute considerations
- Early candidate legacy system architecture evaluation
- Early identification and mitigation of architectural risks



SoS Business / Mission Drivers

Warfare Vignettes
Mission Threads
SoS Architecture Plans

Mission Thread Workshop

System ATAM on candidate legacy system

SoS Architecture Evaluation

SoS Architecture Risks

Problematic systems identified with the augmented mission threads

Augmented Mission Threads
SoS Architecture Challenges

Sys & SW Arch Risks

SoS Architecture
System Architectures

SoS and System Architecture(s) Acquisition / Development

Mission Thread Workshop - Goal

To augment a set of end-to-end System of Systems (SoS) mission threads with quality attribute and engineering considerations with the stakeholders.

To capture at each step of the mission thread AND each SoS quality attribute

- the engineering considerations from diverse stakeholders
- the quality attribute concerns associated with the mission thread
- the applicable use cases for the different nodes and/or systems

To develop technical challenges associated with the threads, and to aggregate the challenges over a number of MTWs

Outputs will inform and drive SoS Architecture Decisions.

Augmentation Process – Per Mission Thread

Two Passes over the Mission Thread:

1) For each event in the mission thread:
- Elicit quality attribute considerations. Capturing any engineering issues, assumptions, challenges, additional use cases and mission threads (with QA context etc.)
- Capture any capability and/or mission issues that arise.

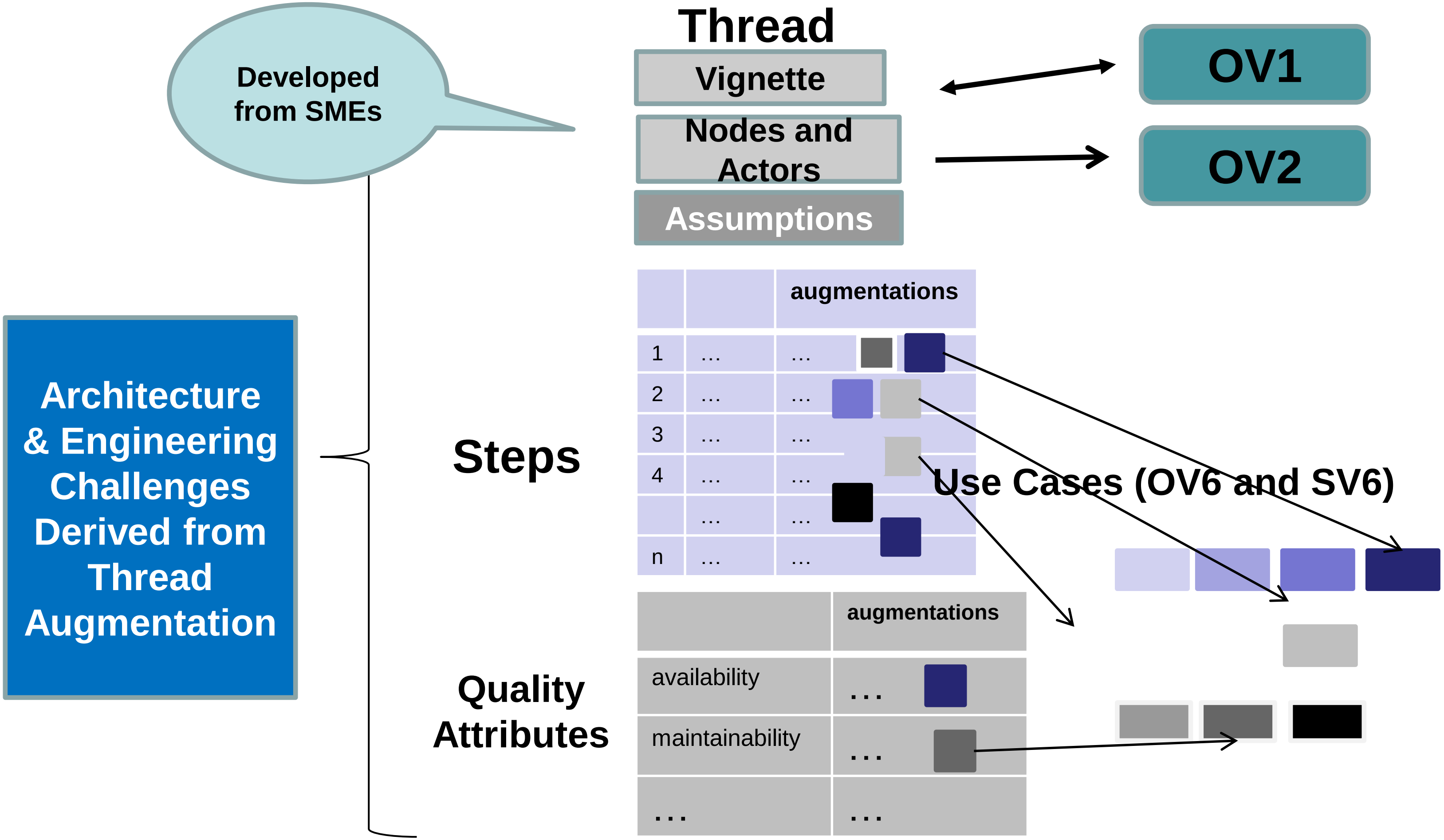
2) For each Quality Attribute - elicit any over-arching quality attribute considerations
- Capturing any over-arching assumptions, engineering issues, challenges, additional use case and mission threads (with QA context) etc.
- Capture any capability and/or mission issues that arise.

Capture any MT extensions for later augmentation

Capture Parking Lot issues – for organization, programmatic, non-technical issues that arise (will not be further pursued in the MTW).

Stakeholder Inputs are Key.

Mission Thread
(augmented via the Mission Thread Workshop)



Developed from SMEs

Thread

Vignette

Nodes and Actors

Assumptions

OV1

OV2

Architecture & Engineering Challenges Derived from Thread Augmentation

Steps

		augmentations
1	…	…
2	…	…
3	…	…
4	…	…
	…	…
n	…	…

Use Cases (OV6 and SV6)

Quality Attributes

	augmentations
availability	…
maintainability	…
…	…

Mission Thread Workshop - Numbers to Date

Client	Description	# MTWs	# Vignettes	# Mission Threads	# of stakeholders
A	IRAD New platform/capability	1	1	2	8
B	New Naval Ship	13	17	37	>200
C	Battle Command	6	3	4	>100
D	Maritime Detection	2	4	4	30
E	NSF	1	3	3	15
F	Air Force Program	1	1	1	10
G	Other Govt Agency	1	2	2	12

Lessons Learned

Preparation Activities

SoS Quality Attributes

Stakeholders

SoS Architecture Development Process

SoS Capabilities and Engineering Considerations

SoS Challenges

Preparation Activities - 1

Development of Mission Threads

Naval Ship had MTWs for each mission area (MA) and inherent capability (IC)

- NOT software related; looking for stakeholder impacts and assumptions that tie the areas together
- Focused on activities and relationships to other assets external to the ship (OV1 is critical)
- Initial leads had trouble building vignettes/threads/QA
  - Needed some coaching and oversight
  - Found good vignette fodder in AoA and DRM documents
- Leads for later workshops attended earlier workshops and developed VERY good vignettes/threads/QA
  - Reviews produced minor changes

Preparation Activities - 2

Architecture Development Strategy (ADS) and Architecture Development Plan (ADP) need to be developed incorporating the architecture-centric approach and identifying how the different products and methods are integrated to develop the architecture

A MTW piloting effort should be performed following the guidance of the ADP and developed architectural process to provide an example for the program with supporting artifacts using operational, developmental and sustainment type threads

A set of architectural quality attributes for the SoS should be defined and vetted with the stakeholders

Training needs to be provided at the program and individual team level

SoS Quality Attributes

Quality Attributes of interest depend on vignette/thread type

- Operational: performance, availability, security, interoperability
- Developmental: legacy reuse, extensibility, openness, integrability
- Sustainment: maintainability, training, deployability, upgradeability

New consideration examples

- Survivability: Machinery MT on how to contain compartmental flooding in a critical compartment resulted in discussion on using new pump technologies to avoid flooding.
- Availability: Machinery MT on failure of a generator has a massive impact on all ship operations and mission
- Availability: Degraded operation on a failure needs to be defined across echelons, and mitigation alternatives defined
- Reduced Manning/Automation

SoS Stakeholders

Evaluation team and customer lead for MTW must get vignette/mission thread/QA/stakeholders ready for the meeting.

Diverse operational experiences eliminate stovepipe mentality

- Discussions on operational misunderstandings, confusions, and gaps were captured

Holding 9 MTWs in 6 weeks with a core team attendance at all provided consistency

Do not mix operational and developmental threads

- They require different stakeholders

Strong third party facilitation allowed operational principles to discuss rather than defend

SoS Architecture Development Process - 1

Each MTW prepared a good OV1 diagram to support the vignette. It was found that a single NR KPP OV1 would not provide the context needed for all teams.

Diagrams developed in PowerPoint or Visio were more than sufficient to support MTW effort, but use of a modeling tool (i.e., System Architect) is probably needed to support development of artifacts in later architectural processes

The MTs developed were the basis for building OV 2,3,5 and SV1, 2 DoDAF diagrams, but additional guidance was needed in architecting process to provide a clear transition

Stakeholders were uncomfortable developing vignettes/mission threads without a CDD-like requirements document

SoS Architecture Development Process - 2

Development of a Mission Thread Description Document (MTDD) is a good way to capture architecture decisions. The MTDD contains the artifacts developed to support the MTWs, outputs of the MTWs (capability gaps, quality attribute augmented mission threads and architectural challenges) and ties to high level use cases.

SoS Architectural Guidelines is needed to provide consistent guidance through the architecting process.

SoS Capabilities and Engineering

High percentage of the data captured was about engineering considerations and gaps

Use cases were identified (and built at later date)

Legacy System Impacts

- One SoS assumed that the legacy systems (sensor, weapon, electrical, mechanical, etc.) would be re-used.
- Another SoS assumed legacy system re-engineering
- Another applied it to extended operational need

SoS Challenges

Each MTW resulted in individual challenges (5-7) for the operation were created by facilitation team and recommended mitigations suggested

- Vetted by the principles
- Led to engineering studies

Where multiple MTWs were held, a set of aggregated challenges were built and mitigations suggested

- Response to a large scale failure, multi-mission planning, global situational awareness, reduced manning, reduced SMEs
- Automated field configuration, Training
- Need to allocate additional time in schedule to gain consensus on aggregate challenges

Legacy System Architecture Evaluation Using Mission Threads as a Starting Point

Mission Threads are the Starting Point

We have used mission thread augmentations to develop system specific scenarios for legacy system architecture evaluation.

Legacy system specific scenarios are developed with the SoS/EA and legacy system stakeholders.

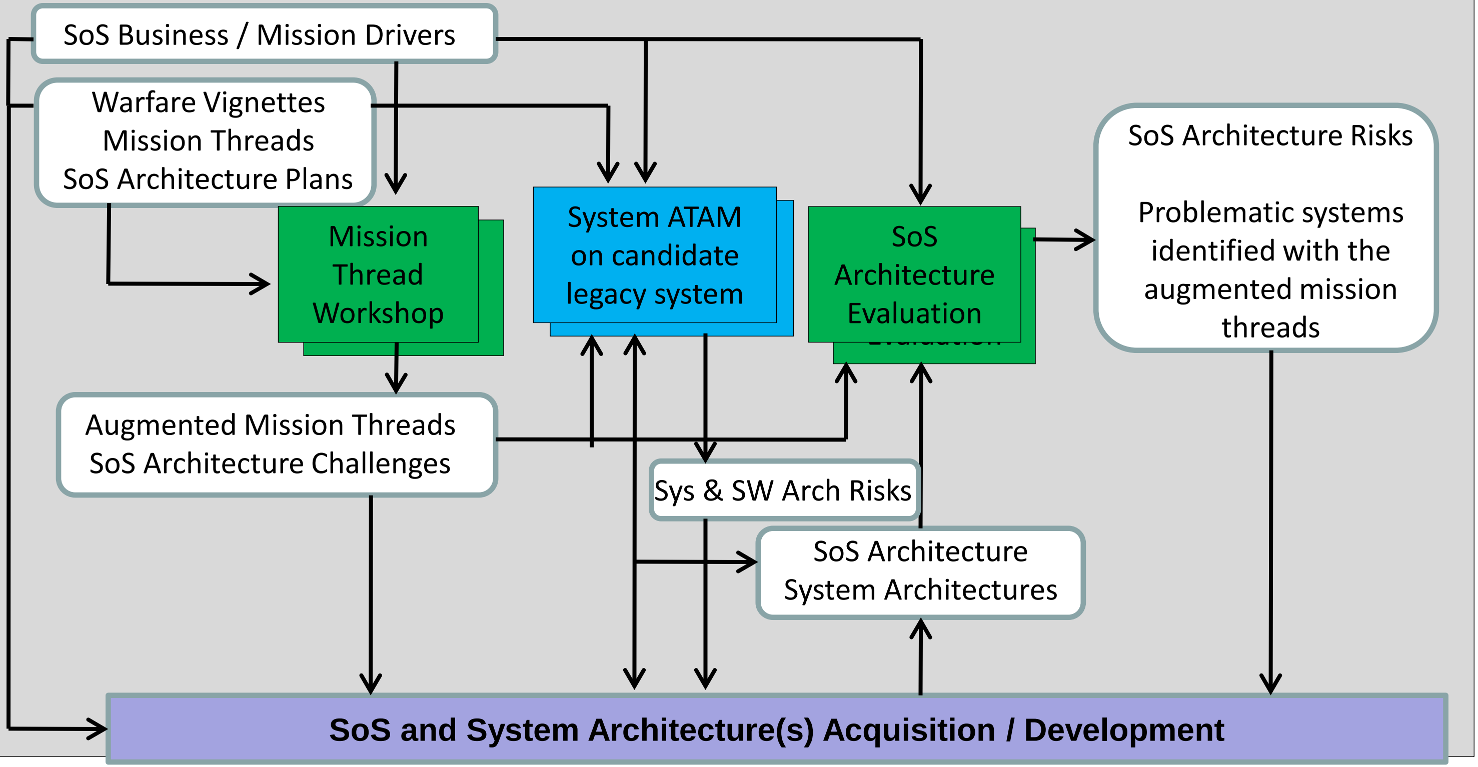
Scenarios are derived from the MT augmentations that pertain to the specific legacy system

- From each MT step augmentation
- From each Quality Attribute augmentation

Scenarios are vetted and prioritized with stakeholders and will be used to evaluate the legacy system & software architecture.

SoS Architecture Quality Attribute Specification and Evaluation Approach

- Early elicitation of quality attribute considerations
- Early candidate legacy system architecture evaluation
- Early identification and mitigation of architectural risks



SoS Business / Mission Drivers
Warfare Vignettes
Mission Threads
SoS Architecture Plans
Mission Thread Workshop
System ATAM on candidate legacy system
SoS Architecture Evaluation
SoS Architecture Risks
Problematic systems identified with the augmented mission threads
Augmented Mission Threads
SoS Architecture Challenges
Sys & SW Arch Risks
SoS Architecture
System Architectures
SoS and System Architecture(s) Acquisition / Development

Example Legacy System Overview - 1

A legacy system was partially modernized

- Exists as a combination of Legacy (75%) and Modernized (25%)
- Exists in two data centers for recovery from catastrophic failure, and under-the-cover data mirroring (one active, one passive)

Modernized portion is:

- COTS based, relational database, service oriented, transactional, batch entries with hundreds of transactions, data warehousing, 24/7/365 operation
- Software is provisioned to processors at configuration time to form round robin load sharing
- Complex interactions between agency users and many participating government agencies and commercial users
- Specialized interfaces to the external users
- Business workflows are implemented using reliable messaging queues between business processes

Example Legacy System Overview - 2

Software architecture mostly undocumented

Data architecture documented in some areas

Original architects no longer on the job

Software maintained by developers, primarily using the software itself and some detailed design documents

Many outstanding software PTRs take a long time to resolve

Limited weekly availability of architects for evaluation

Legacy System Architecture Evaluation - 1

Approach:

- Based on MTW and ATAM concepts
- Developed three end-end business threads, based on the business drivers and elicited from business and operations stakeholders. Representing the three major end-end capabilities of the legacy system.
- Quality attributes were derived from system business drivers and interviewing the architects.
- Augmented the three end-end business threads:
  - Augmenting the steps in the threads; eliciting any legacy system specific scenarios for each step. All quality attribute concerns generated scenarios.
  - for the over-arching quality attributes, developed a legacy system specific utility tree; eliciting concerns and scenarios, using MTW templates
- All of the scenarios are taken together and prioritized (just as in Phase 1 of ATAM).

Legacy System Architecture Evaluation - 2

Approach:

- Evaluation team makeup: 4 SEI evaluators w/ facilitator, 2 Subject Matter Experts from Program Office
- Due to limited weekly availability of architects and lack of documentation, we decided to hold a series of architecture evaluation sessions. Three times a week, two hours per session.
- Decided to focus evaluation sessions on specific topics, e.g., performance, availability, maintainability, etc.
- When all the focused sessions were completed, we executed end-to-end thread sessions.

Legacy System Architecture Evaluation - 3

Results:

- The focused sessions resulted in identifying over 100 architectural risks, 25 non-risks.
- The end-to-end thread sessions resulted in identifying 15 additional risks, mostly dealing with end-to-end issues.
- Seven risk themes were generated. Customer was very satisfied with the results.

Legacy System Architecture Evaluation - 4

Lessons Learned:

- The end-to-end business threads set the proper context for scenario generation for the specific legacy system
- Executing the end-to-end thread sessions last was very beneficial:
  - Put all the previously identified risks into context
  - Helped to focus on end-to-end type risks
  - Helped to understand (and document) the architecture end-to-end
- Lack of a documented architecture was a burden
  - Slowed down the evaluation sessions and extended the schedule. We quickly abandoned the notion of capturing architect’s hand drawings.
  - Never sure if there were places in the architecture that needed further evaluation

Legacy System Architecture Evaluation - 5

Lessons Learned:

- We were satisfied that we had covered the architecture when we finished the end-to-end thread sessions
- We needed 20 evaluation sessions (2 hours each), spanning six weeks. This is not out of line for the total amount of time needed for architects to support an architecture evaluation.
- The architects were very cooperative and open about the process and provided good information, even though they weren’t the original architects.

Contact Information

Mike Gagliardi

Software Engineering Institute
MJG@sei.cmu,edu
412-268-7738

Bill Wood

Software Engineering Institute
WGW@sei.cmu.edu
412-268-7723

Vignettes Are the Starting Point – Example Context